

```
<!DOCTYPE html>
<html>
<body>
```

```
<h1>My First Heading</h1> //head element represented by tags <h1></h1> tags help browser to render data.
```

```
<p>My first paragraph.</p>
```

```
</body>
</html>
```

-
HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading:

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>

HTML paragraphs are defined with the <p> tag:

```
<p>This is a paragraph.</p>
```

HTML links are defined with the <a> tag:
This is a link

HTML images are defined with the tag.

The source file (src), alternative text (alt), width

th, and height are provided as attributes:

```

```


HTML Elements

An HTML element usually consists of a start tag and an end tag, with the content inserted in between:

```
<tagname>Content goes here...</tagname>
```

```
<p>My first paragraph.</p>
```

HTML elements with no content are called empty elements. Empty elements do not have an end tag, such as the `
` element (which indicates a line break).

Nested HTML Elements

HTML elements can be nested (elements can contain elements).

All HTML documents consist of nested HTML elements.

This example contains four HTML elements:

```
<!DOCTYPE html>
```

```
<html>//html element
```

```
<body>// body element
```

```
<h1>My First Heading</h1>//heading element
```

```
<p>My first paragraph.</p>//paragraph element
```

```
</body>
```

```
</html>
```


HTML Attributes

All HTML elements can have attributes

Attributes provide additional information about an element

Attributes are always specified in the start tag
Attributes usually come in name/value pairs like: name="value"

The lang Attribute

The language of the document can be declared in the <html> tag.

The language is declared with the lang attribute.

Declaring a language is important for accessibility applications (screen readers) and search engines :

```
<!DOCTYPE html>
<html lang="en-US">
<body>
```

...

```
</body>
</html>
```

The title Attribute

Here, a title attribute is added to the <p> element. The value of the title attribute will be displayed as a tooltip when you mouse over the paragraph:

```
<p title="I'm a tooltip">
This is a paragraph.
</p>
```

The href Attribute

HTML links are defined with the <a> tag. The link address is specified in the href attribute:

```
<a href="http://www.w3schools.com">This is a link</a>
```

Size Attributes

HTML images are defined with the `` tag.

The filename of the source (`src`), and the size of the image (`width` and `height`) are all provided as attributes:

```

```

The alt Attribute

The alt attribute specifies an alternative text to be used, when an image cannot be displayed.

The value of the attribute can be read by screen readers. This way, someone "listening" to the webpage, e.g. a blind person, can "hear" the element.

```

```

We Suggest: Quote Attribute Values

The HTML5 standard does not require quotes around attribute values.

The href attribute, demonstrated above, can be written as:

```
<a href=http://www.w3schools.com>
```

Single or Double Quotes?

Double quotes around attribute values are the most common in HTML, but single quotes can also be used.

In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes:

```
<p title='John "ShotGun" Nelson'>
```

Or vice versa:

```
<p title="John 'ShotGun' Nelson">
```


HTML Headings

Headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

Note: Browsers automatically add some white space (a margin) before and after a heading.

Note: Use HTML headings for headings only. Don't use headings to make text BIG or bold.

HTML Horizontal Rules

The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The <hr> element is used to separate content (or define a change) in an HTML page:

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```

The HTML <head> Element

The HTML <head> element has nothing to do with HTML headings.

The <head> element is a container for metadata. HTML metadata is data about the HTML document. Metadata is not displayed.

The <head> element is placed between the <html> tag and the <body> tag:

```
!DOCTYPE html>
<html>

<head>
  <title>My First HTML</title>
  <meta charset="UTF-8">
</head>
```

<body>

.
.
.

Note: Metadata typically define the document title, character set, styles, links, scripts, and other meta information.

HTML Paragraphs

The HTML <p> element defines a paragraph

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

With HTML, you cannot change the output by adding extra spaces or extra lines in your HTML code.

The browser will remove any extra spaces and extra lines when the page is displayed:

<p>

This paragraph
contains a lot of lines
in the source code,
but the browser
ignores it.

</p>

<p>

This paragraph
contains a lot of spaces
in the source code,
but the browser
ignores it.

</p>

HTML Line Breaks

The HTML
 element defines a line break.

Use `
` if you want a line break (a new line) without starting a new paragraph:

```
<p>This is<br>a paragraph<br>with line breaks.</p>
```

The HTML `<pre>` Element

The HTML `<pre>` element defines preformatted text.

The text inside a `<pre>` element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks:

```
<pre>
```

```
    My Bonnie lies over the ocean.
```

```
    My Bonnie lies over the sea.
```

```
    My Bonnie lies over the ocean.
```

```
    Oh, bring back my Bonnie to me.
```

```
</pre>
```

```
-----  
-----
```

The HTML Style Attribute

Setting the style of an HTML element, can be done with the style attribute.

The HTML style attribute has the following syntax:

```
<tagname style="property:value;">
```

The property is a CSS property. The value is a CSS value.

HTML Background Color

The background-color property defines the background color for an HTML element.

This example sets the background color for a page to powderblue:

```
<body style="background-color:powderblue;">
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

</body>

HTML Text Color

The color property defines the text color for an HTML element:

```
<h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

HTML Fonts

The font-family property defines the font to be used for an HTML element:

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

HTML Text Size

The font-size property defines the text size for an HTML element:

```
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
```

HTML Text Alignment

The text-align property defines the horizontal text alignment for an HTML element:

```
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```

The <form> Element

The HTML <form> element defines a form that is used to collect user input:

```
<form>
```

.

form elements

.

</form>

An HTML form contains form elements. Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

The <input> Element

The <input> element is the most important form element.

The <input> element can be displayed in several ways, depending on the type attribute.

Here are some examples:

Type	Description
------	-------------

<input type="text">	Defines a one-line text input field
---------------------	-------------------------------------

<input type="radio">	Defines a radio button (for selecting one of many choices)
----------------------	--

<input type="submit">	Defines a submit button (for submitting the form)
-----------------------	---

Text Input

<input type="text"> defines a one-line input field for text input:

Example

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

Radio Button Input

`<input type="radio">` defines a radio button.

Radio buttons let a user select ONE of a limited number of choices:

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

The Submit Button

`<input type="submit">` defines a button for submitting the form data to a form-handler.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's action attribute:

```
<form action="action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse">
<br><br>
  <input type="submit" value="Submit">
</form>
```

The Action Attribute

The action attribute defines the action to be performed when the form is submitted.

Normally, the form data is sent to a web page on the server when the user clicks on the submit button.

In the example above, the form data is sent to a p

age on the server called "action_page.php". This page contains a server-side script that handles the form data:

```
<form action="action_page.php">
```

If the action attribute is omitted, the action is set to the current page.

The Method Attribute

The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data:

```
<form action="action_page.php" method="get">
```

or:

```
<form action="action_page.php" method="post">
```

When to Use GET?

The default method when submitting form data is GET.

However, when GET is used, the submitted form data will be visible in the page address field:

action_page.php?firstname=Mickey&lastname=Mouse

Note: GET must NOT be used when sending sensitive information! GET is best suited for short, non-sensitive, amounts of data, because it has size limitations too.

When to Use POST?

Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field.

POST has no size limitations, and can be used to send large amounts of data.

The Name Attribute

Each input field must have a name attribute to be submitted.

If the name attribute is omitted, the data of that input field will not be sent at all.

This example will only submit the "Last name" input field:

```
<form action="action_page.php">
  First name:<br>
  <input type="text" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse">
<br><br>
  <input type="submit" value="Submit">
</form>
```

Grouping Form Data with <fieldset>

The <fieldset> element is used to group related data in a form.

The <legend> element defines a caption for the <fieldset> element.

```
<form action="action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey"><br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```


The <input> Element

The most important form element is the <input> element.

The `<input>` element can be displayed in several ways, depending on the type attribute.

The `<select>` Element

The `<select>` element defines a drop-down list:

Example:-

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

The `<option>` elements defines an option that can be selected.

By default, the first item in the drop-down list is selected.

To define a pre-selected option, add the `selected` attribute to the option:

```
<option value="fiat" selected>Fiat</option>
```

The `<textarea>` Element

The `<textarea>` element defines a multi-line input field (a text area):

Example:

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

the `rows` attribute specifies the visible number of lines in a text area.

The `cols` attribute specifies the visible width of a text area.

The `<button>` Element

The `<button>` element defines a clickable button:

```
<button type="button" onclick="alert('Hello World!')"
```

```
)">Click Me!</button>
```

This is how the HTML code above will be displayed in a browser:

HTML5 Form Elements

HTML5 added the following form elements:

```
<datalist>
<keygen>
<output>
```

Note: Browsers do not display unknown elements. New elements that are not supported in older browsers will not "destroy" your web page.

HTML5 <datalist> Element

The <datalist> element specifies a list of pre-defined options for an <input> element.

Users will see a drop-down list of the pre-defined options as they input data.

The list attribute of the <input> element, must refer to the id attribute of the <datalist> element.

```
<form action="action_page.php">
  <input list="browsers">
    <datalist id="browsers">
      <option value="Internet Explorer">
      <option value="Firefox">
      <option value="Chrome">
      <option value="Opera">
      <option value="Safari">
    </datalist>
  </form>
```

HTML5 <keygen> Element

The purpose of the <keygen> element is to provide a secure way to authenticate users.

The `<keygen>` element specifies a key-pair generator field in a form.

When the form is submitted, two keys are generated, one private and one public.

The private key is stored locally, and the public key is sent to the server.

The public key could be used to generate a client certificate to authenticate the user in the future

.

Example

A form with a keygen field:

```
<form action="action_page.php">
  Username: <input type="text" name="user">
  Encryption: <keygen name="security">
  <input type="submit">
</form>
```

HTML5 `<output>` Element

The `<output>` element represents the result of a calculation (like one performed by a script).

Example

Perform a calculation and show the result in an `<output>` element:

```
<form action="action_page.php"
  oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  0
  <input type="range" id="a" name="a" value="50">
  100 +
  <input type="number" id="b" name="b" value="50">
  =
  <output name="x" for="a b"></output>
  <br><br>
  <input type="submit">
</form>
```


Input Type Text

`<input type="text">` defines a one-line text input field:

example:

```
<form>
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```

Input Type Password

`<input type="password">` defines a password field:

Example:

```
<form>
  User name:<br>
  <input type="text" name="username"><br>
  User password:<br>
  <input type="password" name="psw">
</form>
```

Input Type Submit

`<input type="submit">` defines a button for submitting form data to a form-handler.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's action attribute:

example:-

```
<form action="action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"
"><br>
```



```
Last name:<br>

```

If you omit the submit button's value attribute, the button will get a default text:

```
<form action="action_page.php">
  First name:<br>
  
```

Input Type Reset

`<input type="reset">` defines a reset button that will reset all form values to their default values:

Example:-

```
<form action="action_page.php">
  First name:<br>
  
```

Input Type Radio

`<input type="radio">` defines a radio button.

Radio buttons let a user select ONLY ONE of a limited number of choices:

example:-

```
<form>
☒
```

```
    <input type="radio" name="gender" value="female">
Female<br>
    <input type="radio" name="gender" value="other">
Other
</form>
```

Input Type Checkbox

`<input type="checkbox">` defines a checkbox.
Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Example:

```
<form>
    <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
    <input type="checkbox" name="vehicle2" value="Car"> I have a car
</form>
```

HTML5 Input Types

HTML5 added several new input types:

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

Input Type Color

The `<input type="color">` is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

```
<form>
  Select your favorite color:
  <input type="color" name="favcolor">
</form>
```

Input Type Date

The `<input type="date">` is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

```
<form>
  Birthday:
  <input type="date" name="bday">
</form>
```

Input Type Datetime-local

The `<input type="datetime-local">` specifies a date and time input field, with no time zone.

Depending on browser support, a date picker can show up in the input field.

```
<form>
  Birthday (date and time):
  <input type="datetime-local" name="bdaytime">
</form>
```

Input Type Email

The `<input type="email">` is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

```
<form>
  E-mail:
  <input type="email" name="email">
```

```
</form>
```

Input Type Month

The `<input type="month">` allows the user to select a month and year.

Depending on browser support, a date picker can show up in the input field.

```
<form>
  Birthday (month and year):
  <input type="month" name="bdaymonth">
</form>
```

Input Type Number

The `<input type="number">` defines a numeric input field.

You can also set restrictions on what numbers are accepted.

The following example displays a numeric input field, where you can enter a value from 1 to 5:

```
<form>
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
</form>
```

Input Type Search

The `<input type="search">` is used for search fields (a search field behaves like a regular text field).

```
<form>
  Search Google:
  <input type="search" name="googlesearch">
</form>
```

Input Type Tel

The `<input type="tel">` is used for input fields that

t should contain a telephone number.

The tel type is currently supported only in Safari 8.

```
<form>
  Telephone:
  <input type="tel" name="usrtel">
</form>
```

Input Type Url

The <input type="url"> is used for input fields that should contain a URL address.

Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

```
<form>
  Add your homepage:
  <input type="url" name="homepage">
</form>
```

Input Type Week

The <input type="week"> allows the user to select a week and year.

Depending on browser support, a date picker can show up in the input field.

```
<form>
  Select a week:
  <input type="week" name="week_year">
</form>
```

